



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
-----------------	-------------	----------------------	---------------------	------------------

10/683,929

10/09/2003

John W. Rapp

1934-13-3

2222

7590 08/26/2008
Bryan A. Santarelli
GRAYBEAL JACKSON HALEY LLP
Suite 350
155 - 108th Avenue NE
Bellevue, WA 98004-5901

EXAMINER

HUISMAN, DAVID J

ART UNIT

PAPER NUMBER

2183

MAIL DATE

DELIVERY MODE

08/26/2008

PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

DETAILED ACTION

1. Claims 1-16, 41-50, and 66-85 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received subsequent to filing and placed of record in the file: Extension of Time, Authorization for Extension of Time for all Replies, and Amendment as received on 5/23/2008, and IDS as received on 6/6/2008.

Information Disclosure Statement

3. It is desirable to avoid the submission of long lists of documents if it can be avoided. Eliminate clearly irrelevant and marginally pertinent cumulative information. If a long list is submitted, highlight those documents which have been specifically brought to applicant's attention and/or are known to be of most significance. See *Penn Yan Boats, Inc. v. Sea Lark Boats, Inc.*, 359 F. Supp. 948, 175 USPQ 260 (S.D. Fla. 1972), *aff'd*, 479 F.2d 1338, 178 USPQ 577 (5th Cir. 1973), *cert. denied*, 414 U.S. 874 (1974). But cf. *Molins PLC v. Textron Inc.*, 48 F.3d 1172, 33 USPQ2d 1823 (Fed. Cir. 1995). See MPEP 2004.
4. An applicant's duty of disclosure of material and information is not satisfied by presenting a patent examiner with “a mountain of largely irrelevant [material] from which he is presumed to have been able, with his expertise and with adequate time, to have found the critical [material]. It ignores the real world conditions under which examiners work.” *Rohm & Haas Co. v. Crystal Chemical Co.*, 722 F.2d 1556, 1573 [220 USPQ 289] (Fed. Cir. 1983), *cert. denied*, 469 U.S. 851 (1984). (Emphasis in original). Patent applicant has a duty not just to

Art Unit: 2183

disclose pertinent prior art references but to make a disclosure in such way as not to “bury” it within other disclosures of less relevant prior art; See *Golden Valley Microwave Foods Inc. v. Weaver Popcorn Co. Inc.*, 24 USPQ2d 1801 (N.D. Ind. 1992); *Molins PLC v. Textron Inc.*, 26 USPQ2d 1889, at 1899 (D.Del 1992); *Penn Yan Boats, Inc. v. Sea Lark Boats, Inc. et al.*, 175 USPQ 260, at 272 (S.D. Fl. 1972).

5. It is impractical for the examiner to thoroughly review each reference, given the number of references cited. By initialing each of the cited references on the accompanying 1449 forms, the examiner is merely acknowledging the submission of the cited references and merely indicating that only a cursory review was made of the cited references.

Claim Objections

6. Claim 1 is objected to because of the following informalities: In line 4, replace “to,” with --to:--. Appropriate correction is required.

7. Claim 6 is objected to because of the following informalities: In line 7, replace “to,” with --to:--. Appropriate correction is required.

8. Claim 7 is objected to because of the following informalities: In line 3, replace “to,” with --to:--. Appropriate correction is required.

9. Claim 8 is objected to because of the following informalities:

- In line 3, replace “comprising,” with --comprising:--.
- In line 5, replace “to,” with --to:--.

Appropriate correction is required.

Art Unit: 2183

10. Claim 9 is objected to because of the following informalities: In line 4, replace “comprising,” with --comprising:--. Appropriate correction is required.
11. Claim 68 is objected to because of the following informalities: In line 4, replace “to,” with --to:--. Appropriate correction is required.
12. Claim 71 is objected to because of the following informalities: In line 3, replace “to,” with --to:--. Appropriate correction is required.
13. Claim 72 is objected to because of the following informalities: In line 3, replace “to,” with --to:--. Appropriate correction is required.
14. Claim 73 is objected to because of the following informalities: In line 2, replace “to,” with --to:--. Appropriate correction is required.
15. Claim 74 is objected to because of the following informalities:
 - In line 4, replace “comprising,” with --comprising:--.
 - In the last line on page 12, replace “to,” with --to:--.Appropriate correction is required.
16. Claim 75 is objected to because of the following informalities:
 - In line 4, replace “comprising,” with --comprising:--.
 - In the 4th to last line on page 13, replace “to,” with --to:--.Appropriate correction is required.
17. Claim 76 is objected to because of the following informalities: In line 4, replace “comprising,” with --comprising:--. Appropriate correction is required.
18. Claim 78 is objected to because of the following informalities:
 - In line 4, replace “comprising,” with --comprising:--.

Art Unit: 2183

- In the 7th and 3rd to last lines of the claim, insert a colon after “to”.

Appropriate correction is required.

19. Claim 79 is objected to because of the following informalities:

- In line 4, replace “comprising,” with --comprising:--.
- In the 3rd to last line of the claim, insert a colon after “to”.

Appropriate correction is required.

20. Claim 80 is objected to because of the following informalities: In line 6, replace “provide” with --providing--. Appropriate correction is required.

Claim Rejections - 35 USC § 102

21. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

22. Claims 6, 41-42, and 80 are rejected under 35 U.S.C. 102(b) as being anticipated by Morikawa.

23. Referring to claim 6, Morikawa has taught a computing machine comprising:

a) a processor operable to broadcast a message that includes data and that includes a header having information indicating a destination of the data. See Fig.10, component 301. The coprocessor receives an instruction via bus 126 and data via bus 124. The instruction and data, collectively, are a “message”. The header is the portion of the message that specifies the destination of the data (which buffer it is stored in upon arrival in the coprocessor). Note that the

Art Unit: 2183

message has this information because it is the message which causes the control circuit to operate, and the control circuit operates the coprocessor.

b) a pipeline accelerator (Fig.10, component 302) coupled to the processor and comprising:

b1) a memory. See Fig.10, and note that input buffer 330/331 and output buffer 332/333 make up a memory (an I/O buffer memory).

b2) a hardwired-pipeline circuit (see Fig.10; the portions of component 302 excluding the buffers; also see the bottom portion of Fig.6 and note that the coprocessor is pipelined) coupled to the memory, including at least one processing pipeline, and operable to:

- receive the message from the processor by receiving the data and the information via at least one same bus line. See Fig.10 and claims 1-2 of Morikawa. The coprocessor receives the “message” over multiple bus lines of the same bus (i.e., the bus comprising lines 124 and 126). Note that a bus is simply a group of wires and any group of wires carrying data/instructions/information may be called a bus.
- extract the data from the message. The message is simply a number of bit inputs into the coprocessor. Among all of the bits that are inputted, the data consumes just a portion of those bits, and is therefore extracted from among all the bits.
- load the extracted data into the memory. Fig.10 shows that the register data coming from the main processor on bus 124 and latch 118 is ultimately put into the memory portion 330/331 by the driver 351/352 of the pipeline circuit.

Art Unit: 2183

- retrieve the extracted data from the memory. Fig.10 shows that the data is retrieved from the memory portion 330/331 by the coprocessor processing unit 211.
- process the retrieved data with a pipeline corresponding to the destination. Once the data has been retrieved from the buffer specified by the destination data, it is processed by the processing pipeline. See column 18, lines 22-27.
- provide the processed data to an external source. See column 18, lines 28-32, and Fig.10. Note that after processing, the data is sent to the processor 301, which is external to the coprocessor (hardwired-pipeline circuit).

24. Referring to claim 41, the method of claim 41 is performed by the circuit of claim 1.

Consequently, claim 41 is rejected for the same reasons set forth in the rejection of claim 1.

Also, the message has information indicating a size of the message. See Fig.2 and note the coprocessor instructions which are part of the message. Each includes an opcode which specifies the size of the immediate data (imm8, imm16, and imm32) of the message.

25. Referring to claim 42, Morikawa has taught a method as described in claim 41.

Furthermore, the method of claim 42 is performed by the circuit of claim 4. Consequently, claim 42 is rejected for the same reasons set forth in the rejection of claim 4.

26. Referring to claim 80, Morikawa has taught the method of claim 41, further comprising:

a) extracting from the header the information indicating the destination of the data. Note that instructions include destinations so that data may be stored.

b) generating from the extracted information an identifier that identifies the pipeline corresponding to the destination. The opcode portion of the instruction specifies which pipeline

Art Unit: 2183

unit will be used in the execution. For instance, a floating-point opcode will denote the use of a floating-point pipeline.

c) storing the identifier in association with the data. The identifier is inherently stored in the instruction register (IR), a cache, main memory, etc.

d) provide the retrieved data to the pipeline in response to the stored identifier. The opcode specifying a particular pipeline will deal with the retrieved data.

27. Claims 41-42 are rejected under 35 U.S.C. 102(b) as being anticipated by Hennessy and Patterson, "Computer Architecture - A Quantitative Approach, 2nd Edition," 1996 (herein referred to as Hennessy).

28. Referring to claim 41, Hennessy has taught a method, comprising:

a) receiving a message that includes data and that includes a header having information indicating a destination of the data and having information indicating a size of the message. See page 155, Fig.3.15, and note that when the pipeline circuit receives the message "LW R1, B" instruction, the header (opcode) specifies that the destination is a register (in this case R1). The message also includes data (the value B). Also, the message has information indicating a size of the message. The "LW" indicates that a word B is to be loaded into R1. Specifically, B is an immediate value that is equivalent in size to a word. The size of B, which is indicated, is a size of the message.

b) extracting the data from the message. The value B is extracted from the message so that it can be stored.

Art Unit: 2183

c) loading the extracted data into a memory. Again, see page 155, Fig.3.15, and note that the “LW R1, B” instruction loads data B into the register file (location R1).

d) retrieving the extracted data from the memory. See page 155, Fig.3.15, and note that when the pipeline executes the “ADD R3, R1, R2” instruction, the data value B, which was previously stored in register R1, is now retrieved from R1 so that it may be processed by the ADD instruction.

e) processing the retrieved data with a hardwired-pipeline circuit that corresponds to the destination of the data. The ADD instruction processes the data value B by adding a value to it. The value added to it is the value stored in register R2 (data value C, which was loaded into R2 by the second LW instruction in Fig.3.15).

f) providing the processed data to an external source. See Fig.3.15 on page 155 and note that after the ADD instruction, the processed data is stored to address A of data memory, which is shown in Fig.3.4 on page 134.

29. Referring to claim 42, Hennessy has taught a method as described in claim 41 wherein providing the processed data comprises:

a) loading the processed data into the memory. See page 155, Fig.3.15 and note that the ADD instruction, after processing the data B (in R1) will load the result into the register file memory at location R3.

b) retrieving the processed data from the memory and providing the retrieved processed data to the external source. See page 155, Fig.3.15, and note that the SW instruction will retrieve the data from register R3 and provide it to the data memory at address A.

Claim Rejections - 35 USC § 103

30. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

31. Claims 9-10, 44-45, 49, and 73 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morikawa.

32. Referring to claim 9, Morikawa has taught a pipeline accelerator comprising:

a) first and second memories. See Fig.10, components 330/331 and 332/333 (input and output buffers).

b) a hardwired-pipeline circuit (see Fig.10; the portions of component 302 excluding the buffers; also see the bottom portion of Fig.6 and note that the coprocessor is pipelined) coupled to the first and second memories and comprising:

b1) an input-data handler operable to receive from an external source a first message that includes raw data and that includes a first header having information indicating a destination of the raw data, to extract the raw data from the message, and to load the raw data into the first memory. See Fig.10, and claims 1-2 of Morikawa. The coprocessor receives an instruction via bus 126 and data via bus 124. The instruction and data, collectively, are a “message”. The header is the portion of the message that specifies the destination of the data (which buffer it is stored in upon arrival in the coprocessor). Note that the message has this information because it is the message which causes the control circuit to operate, and the control circuit operates the coprocessor. The message is simply

Art Unit: 2183

a number of bit inputs into the coprocessor. Among all of the bits that are inputted, the data consumes just a portion of those bits, and is therefore extracted from among all the bits. Furthermore, it should be noted that the portions of the system receiving, controlling receiving, and loading data make up the input data handler.

b2) at least one hardwired pipeline operable to process data. See Fig.10, component 211 and the bottom portion of Fig.6.

b3) a pipeline interface operable to retrieve the raw data from the first memory. See Fig.4, and note that before the data is processed by unit 211, it must be retrieved from the first memory 330/331. The portion of the system retrieving the data would be the pipeline interface.

b4) provide the retrieved raw data to the hardwired pipeline corresponding to the destination. See Fig.4, and note that the data from the first memory 330/331 is provided to the hardwired pipeline 211.

b5) load processed data from the hardwired pipeline into the second memory. See Fig.4, and note that after the data is processed, it is loaded into output buffer 332/333.

b6) an output-data handler operable to retrieve the processed data from the second memory and to provide the processed data to the external source. See column 18, lines 28-32, and Fig.10. The data is retrieved from the memory portion 332/333 and sent to the processor via driver 357/358 and bus 125. The portion of the system controlling the outputting is the output data handler.

b7) Morikawa has not explicitly taught generating a second header having first information indicating a destination of the processed data, generating a second message

Art Unit: 2183

that includes the processed data and the second header, and providing the second message to the external source by providing the processed data and the first information to the external source via at least one same bus line. However, Morikawa has taught sending the data from memory 332/333 to the processor (external source) via bus 357.

Specifically, the data is written to the register file (the software application context) of the processor. One of ordinary skill in the art would've recognized that if data is to be written to a register file, then a destination register must be specified. In this case, either the processor can specify the destination or the coprocessor can specify the destination.

The specifics of such a specification are not disclosed by Morikawa. However, since the coprocessor is performing the operation, the processor can specify the destination as part of a message instead of the processor so that the processor determines the destination only when the coprocessor is finished. This would also allow the processor to save resources for other tasks (not determining the destination). Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa such that the pipeline circuit is operable to generate a message header that includes first information including a destination of the processed data, generate a message that includes the processed data and the header, and provide the message to an external source by providing the processed data and the first information to the external source via at least one same bus line.

33. Referring to claim 10, Morikawa has taught a pipeline accelerator as described in claim 9. Morikawa has further taught that:

Art Unit: 2183

a) the first and second memories each include respective first and second ports. See Fig.10 and note that each of the memories 330/331 and 332/333 include an input and output port.

b) the input-data handler is operable to load the raw data via the first port of the first memory. See Fig.10 and note that data is loaded into first memory 330/331 via its input port.

c) the pipeline interface is operable to retrieve the raw data via the second port of the first memory and to load the processed data via the first port of the second memory. See Fig.10 and note that the data is read from memory 330/331 via its output port, it is processed by component 211, and then the processed data is written to the second memory 332/333 via its input port.

d) the output-data handler is operable to retrieve the processed data via the second port of the second memory. See Fig.10 and note that when the data is to be sent to the processor 301, it is first retrieved from memory 332/333 via its output port.

34. Referring to claim 44, the method of claim 44 is performed by the circuit of claim 9. Consequently, claim 44 is rejected for the same reasons set forth in the rejection of claim 9.

35. Referring to claim 45, Morikawa has taught a method as described in claim 44. Furthermore, the method of claim 45 is performed by the circuit of claim 10. Consequently, claim 45 is rejected for the same reasons set forth in the rejection of claim 10.

36. Referring to claim 49, Morikawa has taught a method as described in claim 44. Furthermore, Morikawa has inherently taught setting parameters for loading and retrieving the raw data, processing the retrieved data, and loading and providing the processed data. That is, if the system is going to load data from the processor, process the data, and then provide data back to the processor, then the system must set which data will be loaded (i.e., which register the data will be coming from), it must set which operation is to be performed on the data, and it must set

Art Unit: 2183

which register will be receiving the processed data result. Without these parameters, the functions could not be performed.

37. Referring to claim 73, Morikawa has taught a pipeline accelerator as described in claim 9, wherein the input-data handler is further operable to:

- a) extract from the header the information indicating the destination of the data. Note that instructions include destinations so that data may be stored.
- b) generate from the extracted information an identifier that identifies the pipeline corresponding to the destination. The opcode portion of the instruction specifies which pipeline unit will be used in the execution. For instance, a floating-point opcode will denote the use of a floating-point pipeline.
- c) store the identifier in association with the data. The identifier is inherently stored in the instruction register (IR), a cache, main memory, etc.
- d) wherein the pipeline interface is further operable to provide the retrieved data to the pipeline in response to the stored identifier. The opcode specifying a particular pipeline will deal with the retrieved data.

38. Claims 11 and 46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morikawa in view of Fette.

39. Referring to claim 11, Morikawa has taught a pipeline accelerator as described in claim 9. Morikawa has not taught a third memory coupled to the hardwired-pipeline circuit, wherein the hardwired pipeline is operable to generate intermediate data while processing the raw data, and wherein the pipeline interface is operable to load the intermediate data into the third memory and

Art Unit: 2183

to retrieve the intermediate data from the third memory. However, Fette has taught such a concept. Specifically, Fette has taught that coprocessors contain multiply-accumulate circuitry. See the last sentence of the abstract. A multiply-accumulator is known to include an accumulation register (third memory). With a multiply-accumulate (MAC) operation, values (raw data) are multiplied to generate a multiplication result. This result is then added to the value already in the accumulation register to generate intermediate data. This intermediate data is then stored in the third memory (back into the accumulator), where it is later retrieved and added to the next multiplication result. A MAC operation is a common operation, and it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa such that the coprocessor performs MAC operations and therefore includes the claimed third memory and its purpose. One would have been motivated to make such a combination to increase the functionality of the coprocessor by giving it MAC functionality.

40. Referring to claim 46, Morikawa has taught a method as described in claim 44. Furthermore, the method of claim 46 is performed by the circuit of claim 11. Consequently, claim 46 is rejected for the same reasons set forth in the rejection of claim 11.

41. Claims 12 and 15 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morikawa in view of FOLDOC.

42. Referring to claim 12, Morikawa has taught a pipeline accelerator as described in claim 9. a) Morikawa has not taught that the first and second memories are respectively disposed on first and second integrated circuits. However, as shown in Nerwin v. Erlichman 168 USPQ 177 (1969), to make separable is generally not given patentable weight or would have been an

Art Unit: 2183

obvious improvement. In this case, it would have been obvious to have the buffers of the coprocessor of Fig.10 to be on separate chips. For instance, everything prior to the processing unit 211 would be on one chip, while the rest would be on another. A person of ordinary skill in the art would have recognized that separating the components into multiple chips allows for more flexibility and cost savings. For example, if the coprocessor were divided into multiple chips and one chip is malfunctioning or is defective, just that chip would need to be replaced while the other functioning chip would be left in tact. This results in not having to spend money replacing the functioning chip. Another advantage would be to gain the ability to upgrade a portion of the coprocessor. As a result, in order to increase flexibility, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa to include the first and second memories on separate integrated circuits.

b) Morikawa has not taught that the pipeline circuit is disposed on a field-programmable gate array (FPGA). However, FOLDOC has taught that FPGAs are devices which are programmed after manufacture time. See the 1st paragraph of FOLDOC. This is clearly an advantage as designers are able to reprogram/reconfigure the device after it has been made. That is, if any new functionality is desired, a new chip would not have to be manufactured. Instead, the associated functionality would simply be programmed into the FPGA. Consequently, in order to make the pipeline circuit of Morikawa reconfigurable and, consequently, more flexible, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa such that the pipeline is disposed on an FPGA, as taught by FOLDOC.

43. Referring to claim 15, Morikawa has taught a pipeline accelerator as described in claim 9.

a) Morikawa has further taught that each of the input-data handler, hardwired pipeline, pipeline

Art Unit: 2183

interface, and output-data handler has a respective operating configuration. That is, each component in Morikawa operates in a specific fashion and consequently, each component inherently has a respective operating configuration.

b) Morikawa has not taught a configuration manager coupled to and operable to set the operating configurations of the input-data handler, hardwired pipeline, pipeline interface, and output-data handler. However, FOLDOC has taught that FPGAs are devices which are programmed after manufacture time. See the 1st paragraph of FOLDOC. This is clearly an advantage as designers are able to reprogram/reconfigure the device after it has been made. That is, if any new functionality is desired, a new chip would not have to be manufactured. Instead, the associated functionality would simply be programmed into the FPGA. Consequently, in order to make the pipeline circuit of Morikawa reconfigurable and, consequently, more flexible, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa such that the system is disposed on an FPGA. Furthermore, an FPGA would be coupled to a configuration manager so that the FPGA may be programmed to include the desired functionality.

44. Claims 13-14 and 47-48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morikawa in view of Frey et al., U.S. Patent No. 5,185,871 (herein referred to as Frey).

45. Referring to claim 13, Morikawa has taught a pipeline accelerator as described in claim 9. Morikawa has not taught an input-data queue coupled to the input-data handler and the pipeline interface, wherein the input-data handler is operable to load into the input-data queue a pointer to a location of the raw data within the first memory and wherein the pipeline interface is operable

Art Unit: 2183

to retrieve the raw data from the location using the pointer. However, Frey has taught such a concept. See Fig.3, components 17 and 21, and column 11, line 67, to column 12, line 5.

Essentially, addresses of operands in a multi-operand buffer are stored in an operand fetch queue so that when it is time to fetch the operands, they are located appropriately. A person of ordinary skill in the art would have recognized that by implementing a multi-operand buffer in Morikawa (for buffer 330/331), multiple operands would be buffered at a time, which ensures that the coprocessor always has enough work to do. Also, if the coprocessor were to stall in any way, the multi-operand buffer would still be able to buffer operands from the processor, thereby allowing the processor to continue on with other work. With such a system, saving the pointer of an operand allows for the location of the operand in the buffer. Clearly, when an instruction is to operate on an operand, the correct operand should be located, and so the location of the operand must be remembered. Therefore, in order to buffer more operands, ensure work, and avoid stalls, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa to include a multi-operand input buffer and an input-data queue for holding a pointer to an operand (raw data) within the buffer and then using the pointer to locate the desired operand.

46. Referring to claim 14, Morikawa has taught a pipeline accelerator as described in claim 9. Morikawa has not taught an output-data queue coupled to the output-data handler and the pipeline interface wherein the pipeline interface is operable to load into the output-data queue a pointer to a location of the processed data within the second memory and wherein the output-data handler is operable to retrieve the processed data from the location using the pointer. However, Frey has taught keeping a queue of pointers so that data in a multiple-entry buffer

Art Unit: 2183

would be appropriately located. See Fig.3, components 17 and 21, and column 11, line 67, to column 12, line 5. A person of ordinary skill in the art would have recognized that by implementing a multi-result buffer in Morikawa (for buffer 332/333), multiple results would be buffered at a time, which ensures that the coprocessor would function smoothly if bus contention were present. For instance, the regular processor may be trying to write to its own register file. If this is the case, the coprocessor cannot write a result to the register file at the same time. Consequently, if only a single entry output buffer exists, then the result must be stored in that buffer until the register file is available. Until it's available, another result cannot be produced as the previous result is already in the buffer. This would result in stalling the coprocessor. With a multiple-entry buffer, the coprocessor would be able to continue executing and storing results until the register file becomes available. And, with such a system, saving the pointer of the result allows for the location of the result in the buffer. Clearly, when a result is to be written to the register file, the correct result should be written, and so the location of the result must be remembered. Therefore, in order to avoid stall potential, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa to include a multiple-entry output buffer and an output-data queue for holding a pointer to a result (processed data) within the buffer and then using the pointer to locate the desired data.

47. Referring to claim 47, Morikawa has taught a method as described in claim 44. Furthermore, the method of claim 47 is performed by the circuit of claim 13. Consequently, claim 47 is rejected for the same reasons set forth in the rejection of claim 13.

Art Unit: 2183

48. Referring to claim 48, Morikawa has taught a method as described in claim 44. Furthermore, the method of claim 48 is performed by the circuit of claim 14. Consequently, claim 48 is rejected for the same reasons set forth in the rejection of claim 14.

49. Claims 16 and 50 are rejected under 35 U.S.C. 103(a) as being unpatentable over Morikawa.

50. Referring to claim 16, Morikawa has taught a method as described in claim 9.

a) Morikawa has inherently taught that each of the input-data handler, hardwired pipeline, pipeline interface, and output-data handler has a respective operating status. That is, at the very least, each component in the system is either operating or not operating (and these are statuses).

b) Morikawa has not taught an exception manager coupled to and operable to identify an exception in the input-data handler, hardwired pipeline, pipeline interface, or output-data handler in response to the operating statuses. However, Official Notice is taken that checking for errors in a pipeline during processing is well known and accepted in the art. For instance, if a coprocessor is performing a divide operation, which is a well known operation performed by coprocessors, then a division of a number by 0 should raise an exception, as it is an illegal operation. This type of error should be monitored so that the system can take appropriate action to correct the error. As a result, in order to ensure proper execution, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa such that the coprocessor of Morikawa performs divide operations (in order to obtain the ability to perform division) and exceptions are detected in a hardwired pipeline.

Art Unit: 2183

51. Referring to claim 50, Morikawa has taught a method as described in claim 44. While Morikawa has not explicitly taught determining whether an error occurs during the loading and retrieving of the raw data, the processing of the retrieved data, and the loading and providing of the processed data, Official Notice is taken that checking for errors during processing is well known and accepted in the art. For instance, if a coprocessor is performing a divide operation, which is a well known operation performed by coprocessors, then a division of a number by 0 should raise an error as it is an illegal operation. This type of error should be monitored so that the system can take appropriate action to correct the error. As a result, in order to ensure proper execution, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Morikawa such that the coprocessor of Morikawa performs divide operations (in order to obtain the ability to perform division) and errors are detected during the loading and retrieving of the raw data, the processing of the retrieved data, and the loading and providing of the processed data.

Allowable Subject Matter

52. Claims 1-5, 7-8, 43, 66-72, 74-79, and 81-85 are allowed. Please correct any objections associated with these claims.

Response to Arguments

53. Applicant's arguments filed on May 23, 2008, have been fully considered but they are not persuasive.

Art Unit: 2183

54. Applicant argues the novelty/rejection of claim 6 on page 22 of the remarks, in substance that:

“Claim 6 as amended recites a processor operable to broadcast a message that includes data and that includes a header having information indicating a destination of the data, and a hardwired-pipeline circuit operable to receive the message from the processor by receiving the data and the information via at least one same bus line.”

55. These arguments are not found persuasive for the following reasons:

a) The examiner asserts that Morikawa can be viewed as having a message (same) bus that comprises sub-buses 124 and 126. The message bus, therefore, would be a single bus having at least one bus line for transferring a message. Note from the Microsoft Computer Dictionary, 5th Edition (which is cited as extrinsic evidence), that bus is merely defined as “a set of hardware lines used for data transfer among the components of a computer system. A bus is essentially a shared highway that connects different parts of the system...”. Consequently, it can be seen that the set of wires 124 and 126 for a bus.

56. Applicant argues the novelty/rejection of claim 41 on page 22 of the remarks, in substance that:

“Morikawa does not disclose receiving a message that includes a header having information indicating a size of the message. Referring to sections 11 and 77 of the office action and to Morikawa's FIG. 2, the Examiner states that “the instruction opcode indicates an immediate value size (imm8, imm16, and imm32) of the message.” The immediate values imm8, imm16, and imm32 are immediate data that are expanded and then multiplied by another number in the source register Dn to generate a result (col. 7, lines 35- 40). Consequently, contrary to the Examiner's assertion, these immediate values do not indicate the size of the instruction opcode or of any portion the instruction opcode.”

57. These arguments are not found persuasive for the following reasons:

a) Whether data is expanded and then used in an operation or not, the opcode still indicates immediate value size (and hence, message size).

Art Unit: 2183

58. Applicant argues the novelty/rejection of claim 41 on pages 23-24 of the remarks, in substance that:

“Hennessy does not disclose receiving a message that includes data and that includes a header having information indicating a destination of the data and information indicating a size of the message. Referring to Hennessy's p. 155, contrary to the Examiner's position, the value B does not indicate a size of a message. First, B itself cannot be a message, because it does not include information indicating a destination of itself (the Examiner's position is that B corresponds to "data" recited in claim 41). Second, according to the Examiner's position, B is a value to be loaded into the destination register R1. But even if the combination of B and R1 compose a message, there is no indication that B indicates the combined size of B and the binary value identifying R1.”

59. These arguments are not found persuasive for the following reasons:

a) As discussed in the rejection of claim 41 above, the message has information indicating a size of the message. The “LW” indicates that a word B is to be loaded into R1. Specifically, B is an immediate value that is equivalent in size to a word. The size of B, which is indicated, is a size of the message.

60. The argument with respect to claim 9 is responded to in the same manner that the argument with respect to claim 6 was responded to above.

Conclusion

61. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

Art Unit: 2183

MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

62. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Applicant is reminded that in amending in response to a rejection of claims, the patentable novelty must be clearly shown in view of the state of the art disclosed by the references cited and the objections made. Applicant must also show how the amendments avoid such references and objections. See 37 CFR § 1.111(c).

Loewenthal et al., U.S. Patent No. 5,712,922, has taught a neural network accelerator that receives data from a preprocessor, stores the data in a memory, processes the data without executing an instruction, and stores the result in a memory for access by the CPU.

Freitag, Jr. U.S. Patent No. 6,237,054, has taught a processor coupled to multiple configuration logic blocks that may be configured to perform a logic operation on data and output a result to the processor.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to DAVID J. HUISMAN whose telephone number is (571)272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

Art Unit: 2183

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/David J. Huisman/
Primary Examiner, Art Unit 2183
August 19, 2008